REMARKS

Applicants appreciate the continued thorough examination of the present application that is evidenced in the Official Action of September 13, 2006 (the "Office Action"). Applicants respectfully request reconsideration of the rejections set forth therein for the reasons provided below.

In the Office Action, Claims 1-12, 14, 16-18, 20 and 22-39 were rejected as unpatentable over U.S. Patent No. 6,141,705 to Anand et al. ("Anand") in view of U.S. Patent Publication No. 2003/0014623 to Freed et al. ("Freed"). Office Action, p. 2. Applicants respectfully submit that Freed and Anand do not teach or suggest each and every recitation of the Claims, either alone or in combination. To the contrary, Anand explicitly teaches away from many recitations of the claims, as explained below.

Claim 1 recites as follows (emphasis added):

> 1. A method of performing security processing in a computing network comprising a local unit having an operating system kernel executing at least one application program, comprising:
>
> receiving a first request **at the operating system kernel** from the application program to initiate a communication with a remote unit;
>
> providing a second request **from the operating system kernel** to a security offload component which performs security handshake processing, the second request directing the security offload component to secure the communication with the remote unit; and
>
> providing a control function **in the operating system kernel** for initiating operation of the security handshake processing by the security offload component.

Accordingly, Claim 1 is directed to a method of performing security processing in a computing network in which operation of security handshake processing by a security offload component is initiated by a control function in an operating system kernel. As explained in the present application, such operations may be performed transparently by the operating system kernel on behalf of an application program that may have no awareness of security processing. See Application, p. 7, ll. 17-18 ("An object of the present invention is to enable security processing to be transparent to application code."). See also, Application p. 19, l. 20 to p. 20, l. 3 ("This offloading technique may be used with any of the application scenarios previously

described (including those that issue SSL directives from an application and those that issue SSL calls which operate as no-ops), for applications that are SSL-enabled, SSL-aware, or neither SSL-enabled nor SSL-aware.").

In distinct contrast, Anand is directed to a system by which security processing is performed by an offload component (e.g. a NIC) under the direction and control of an application program. As explained in Anand, "[a]n **application executing on the computer system** first queries the processing, or task offload capabilities of the NIC, and then selectively enables those capabilities that may be subsequently needed by the application." Anand, Abstract. Furthermore, Anand states that "[o]nce an **application** has discerned the capabilities of a particular NIC, it will selectively utilize any of the enabled task offload capabilities of the NIC by appending packet extension data to the network data packet that is forwarded to the NIC." Anand, Abstract (emphasis added). Thus, Anand explicitly requires the application program to query the capabilities of an offload processor and then selectively enable those capabilities.

The Office Action cites Figure 3 and the accompanying text of Anand at col. 10, ll. 27-47 as disclosing the claimed steps of receiving a first request at the operating system kernel from the application program to initiate a communication with a remote unit, providing a second request from the operating system kernel to a security offload component directing the security offload component to secure the communication with the remote unit, and providing a control function in the operating system kernel for initiating operation of the security handshake processing by the security offload component. Office Action, pp.2-3.

However, Anand Figure 3 does not teach or suggest kernel-based offload security processing as recited in Claim 1. In fact, the system of Anand Figure 3 expressly **excludes** kernel-based offload security processing as recited in Claim 1. For example, the Office Action cites element 128 of Anand Figure 3 as teaching the step of providing a second request from the operating system kernel to a security offload component directing the security offload component to secure the communication with the remote unit. However, element 128 of Anand Figure 3 is explicitly labeled a "Transport Protocol ***Driver***" which a skilled person would necessarily understand to be different from an operating system kernel.

The Office Action further cites element 100 of Anand Figure 3 as providing the claimed step of providing a control function in the operating system kernel for initiating operation of the security handshake processing by the security offload component. Office Action, p. 3. However, element 100 of Anand Figure 3 is simply the NIC, which in the system of Anand communicates with a Network ***Driver*** 116, not with an operating system kernel. In fact, Anand expressly differentiates the network drivers from the operating system. See Anand, col. 8, ll. 26-31 ("Each driver, typically implemented as a software component provided by the vendor of the corresponding NIC, is responsible for sending and receiving packets over its corresponding network connection and for managing the NIC on behalf of the operating system.").

The Office Action cites element 140 of Anand Figure 3 as teaching the step of receiving a first request at the operating system kernel from the application program to initiate a communication with a remote unit. Office Action, p. 2. However, element 140 is explicitly labeled "Application Data" in Anand Figure 3, which does not imply that the application program that generated the application data does not implement other functions/blocks shown in Anand Figure 3. Indeed, Anand specifically states, as explained above, that the application program is required to query the capabilities of an offload processor and then selectively enable those capabilities. Applicant notes that Anand Figure 3 merely provides a logical description of the flow of application data through the layers of a networking model. See Anand, col. 10, ll. 13-16 ("By way of example, FIG. 3 illustrates the path followed by the packet as it proceeds down through the respective layers to the NIC...").

The Office Action cites Freed as disclosing a method for secure communications between a client and a server including managing a communication negotiation between the client and the server. Office Action, p. 3. Thus, Freed is not cited as providing the missing recitations noted above, and even if combined, Freed and Anand would not teach every element of Claim 1.

Similarly, Independent Claim 33 recites (emphasis added):

> 33. A method of performing security processing in a computing network including a local unit having an operating system kernel executing at least one application program, comprising:

providing a security offload component which performs security session establishment and control processing;

providing a control function **in the operating system kernel** for initiating operation of the security session establishment and control processing by the security offload component;

receiving a request **at the operating system kernel** from the application program to initiate a communication with a remote unit; and

directing the security offload component to secure the communication with the remote unit **in response to the request**.

For at least the reasons discussed above, Applicants respectfully submit that Anand does not teach or suggest providing a control function in an operating system kernel for initiating operation of security session establishment and control processing by a security offload component, receiving a request at the operating system kernel from the application program to initiate a communication with a remote unit, or directing the security offload component to secure the communication with the remote unit in response to the request, as recited in Claim 33. The Office Action cites Freed as disclosing a method for secure communications between a client and a server including managing a communication negotiation between the client and the server. Office Action, p. 8. Thus, Freed is not cited as providing the missing recitations noted above, and even if combined, Freed and Anand would not teach every element of Claim 33.

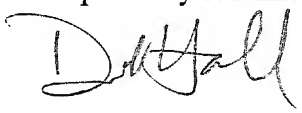
Independent Claims 34-35 are system and computer program product claims containing recitations similar to those of Independent Claim 33, and are submitted to be patentable for similar reasons as Claim 33. Accordingly, Applicants respectfully submit that Independent Claims 1 and 33-35 are patentable over Anand and Freed, alone or in combination.

The dependent Claims are patentable at least as being dependent upon patentable base claims.

CONCLUSION

In light of the above remarks, Applicants respectfully submit that the above-entitled application is in condition for allowance. Favorable reconsideration of this application is respectfully requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (919) 854-1400.

Respectfully submitted,



David C. Hall
Registration No. 38,904

Myers Bigel Sibley & Sajovec, P.A.
P.O. Box 37428
Raleigh, NC 27627
919-854-1400
919-854-1401 (Fax)